

NAME

Siddhant Iyer

DATE

28th January 2026

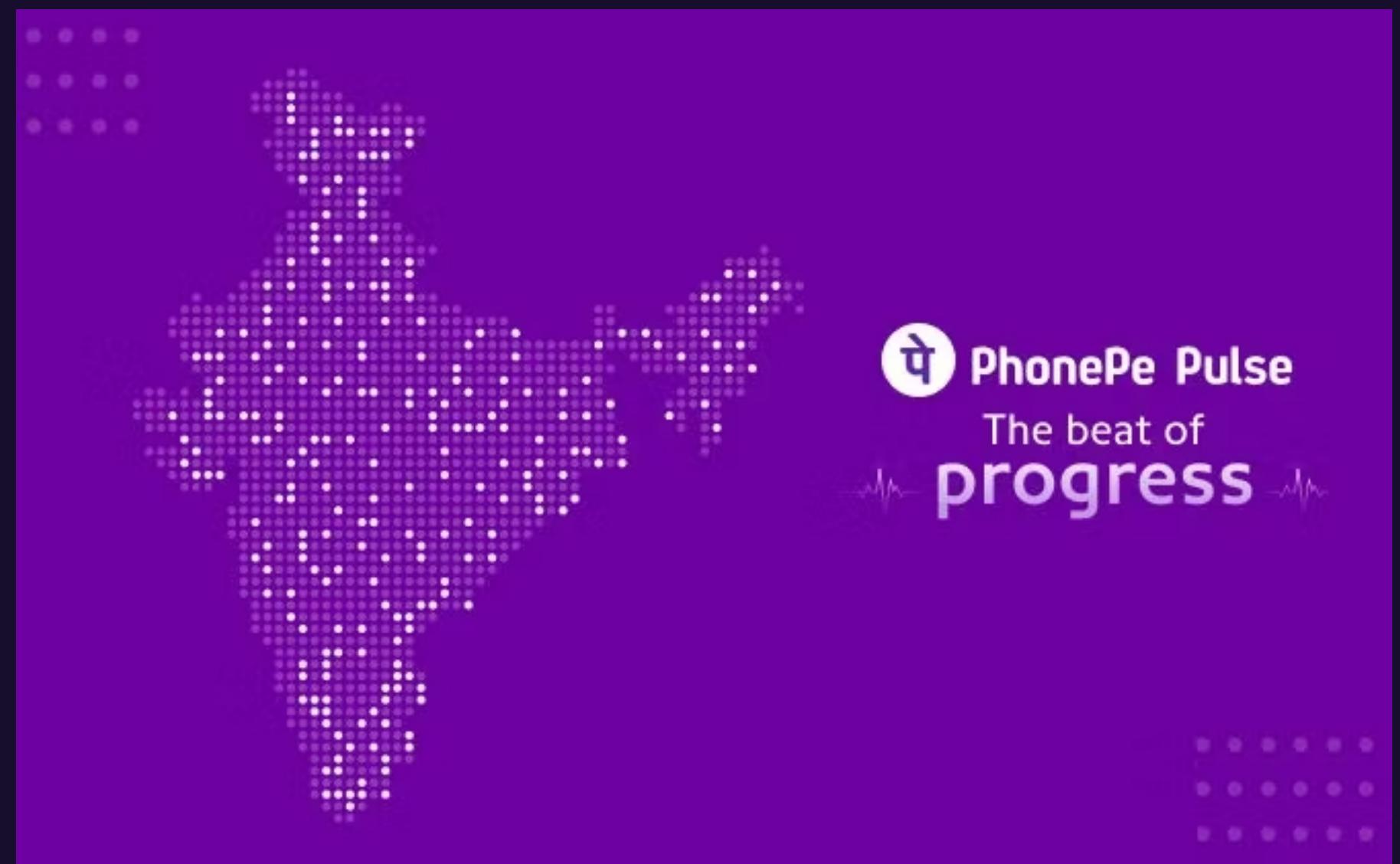


# PhonePe Pulse

## Business Analytics Project



# Introducing PhonePe Pulse



**Leading digital payment  
platform in India**

# Digital Payments in India

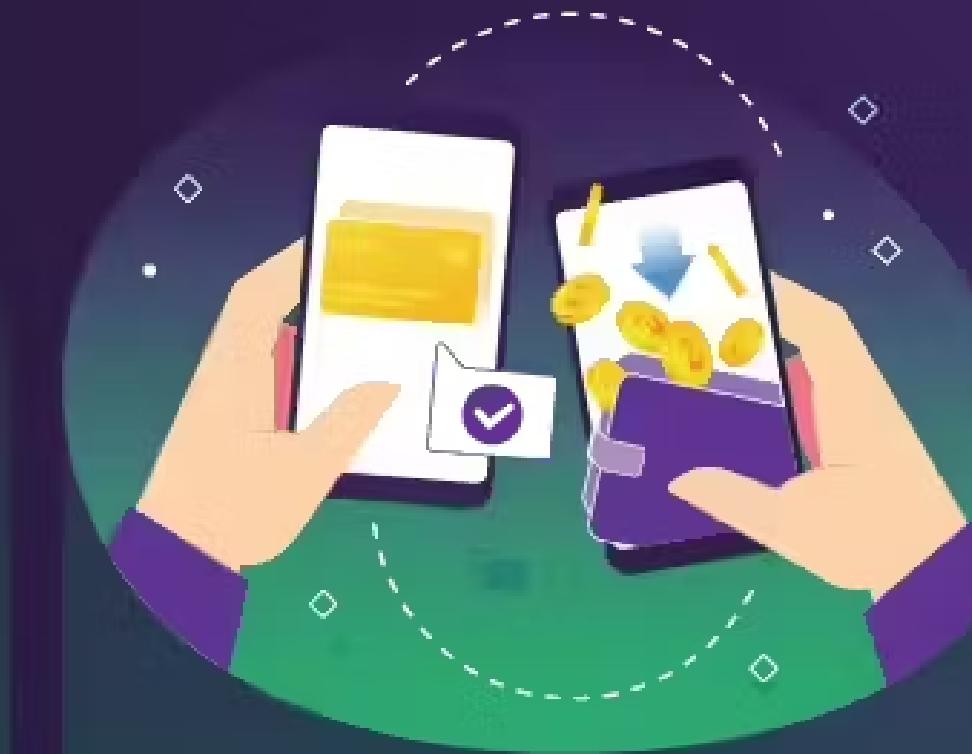
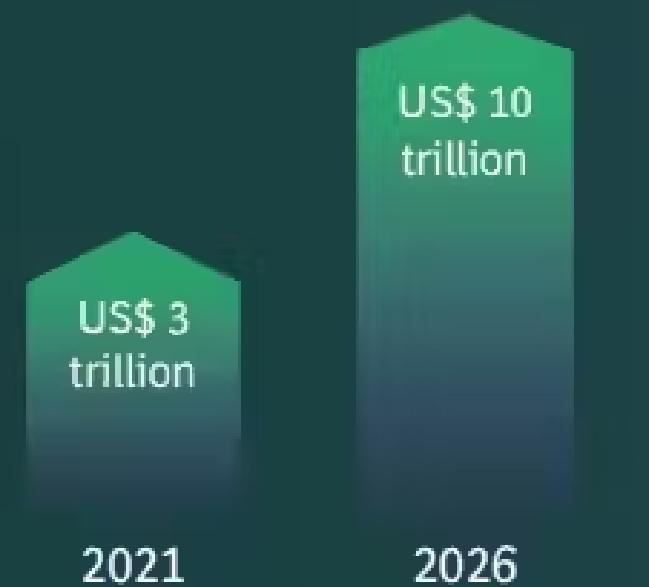
A US\$10 Trillion Opportunity

BCG  
25 YEARS IN INDIA  
LEADING WITH  
IMPACT  
INNOVATION  
INTEGRITY

PhonePe Pulse



>3X  
growth expected in India's  
digital payments

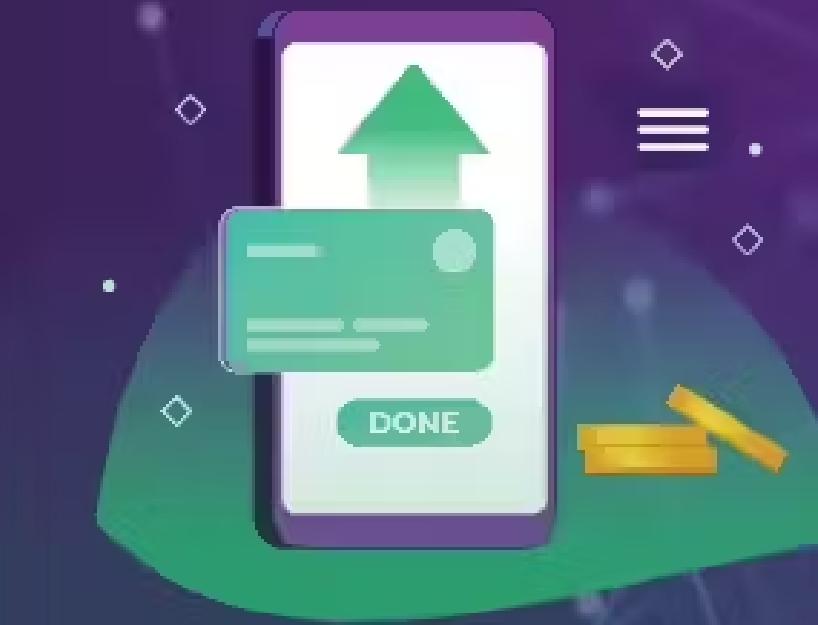


India to become a **digital  
payment economy**

40%      65%



Share of digital  
payment transactions



Digital merchant payments  
to be major driver of growth

20%      65%



Share of digital in total  
merchant payments

# Project Overview

With an increase in reliance of digital payment systems like PhonePe, to improve services and target users effectively, it becomes crucial to understand:

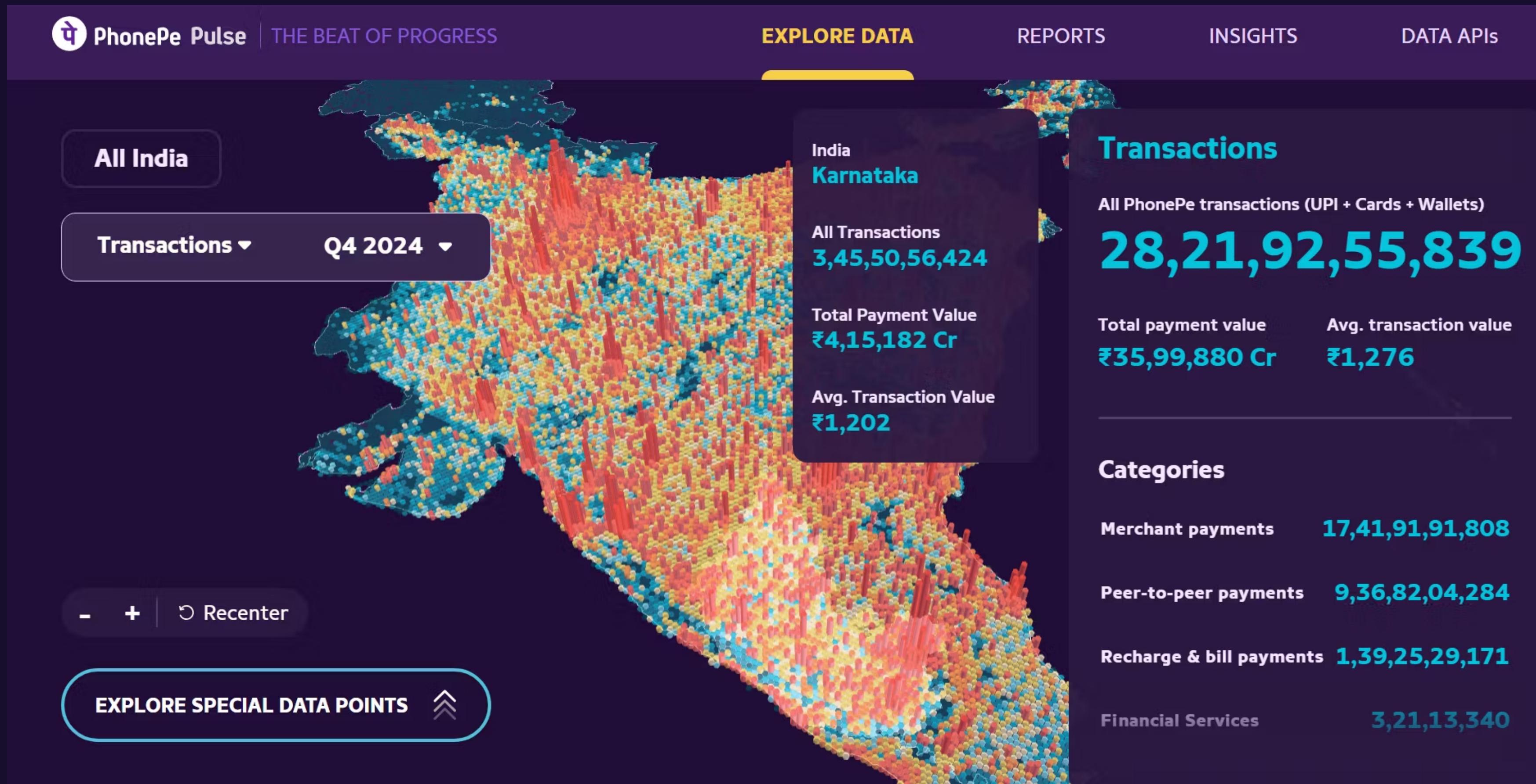
- **Transaction dynamics**
- **User engagement**
- **Insurance-related data**

The project therefore aims to build an interactive multi-page PhonePe Pulse Streamlit dashboard to:

1. Analyze and **visualize** data for three domains: **Transactions, Insurance, and Users**
2. Display a **choropleth map** for state-wise data exploration
3. Tackle five **business case studies** and suggest **recommendations** as appropriate to improve growth.



# PhonePe Pulse Dashboard Design Inspiration



# Core Tools and Deployment Stack

## CORE DEVELOPMENT STACK

**Frontend:** Streamlit (*Interactive dashboard + Plotly charts + Chloropleth map*).

**Backend:** SQL (MySQL using PyMySQL, SQLAlchemy), Python (Jupyter notebook), HTML, CSS (*dashboard styling*).

**Database:** Local Host Environment (*Testing*), Railway MySQL (*Production Hosting*)

## DEPLOYMENT INFRASTRUCTURE

**Version Control:** GitHub (*Public repos + auto-deploys*)

**Database CLI:** Railway CLI (*Railway runs MySQL queries in the background*).

**Secrets Management:** Streamlit Secrets.toml (*Railway DB Credentials*)

**Hosting:** Streamlit Community Cloud (*Zero-cost production*)



# End-to-End Project Pipeline

THE JOURNEY FROM EXTRACTING DATA TO HOSTING THE INTERACTIVE DASHBOARD LIVE

- 
- |    |   |    |   |
|----|---|----|---|
| 01 | <b>Cloning GitHub Repository</b> in Python and Importing Data | 05 | Querying The Database and Visualizing Results                           |
| 02 | Extracting Data From <b>Nested JSON File</b>                  | 06 | Interactive <b>Multi-Page Streamlit Dashboard</b> Creation              |
| 03 | Converting Data into <b>CSV Files</b>                         | 07 | Setting Up The Online <b>MySQL Database</b> On <b>Railway</b>           |
| 04 | SQL Database and Table Creation Using This Data               | 08 | Hosting The Dashboard Live On <b>Streamlit Community Cloud</b> Platform |
-

# Components of the Dashboard

## ABOUT

- What can you find in this dashboard?
- Executive summary describing PhonePe Pulse, recent trends in digital payments across India, potential for growth, the need to target undeserved regions and issues faced by payment providers.

## EXPLORE DATA

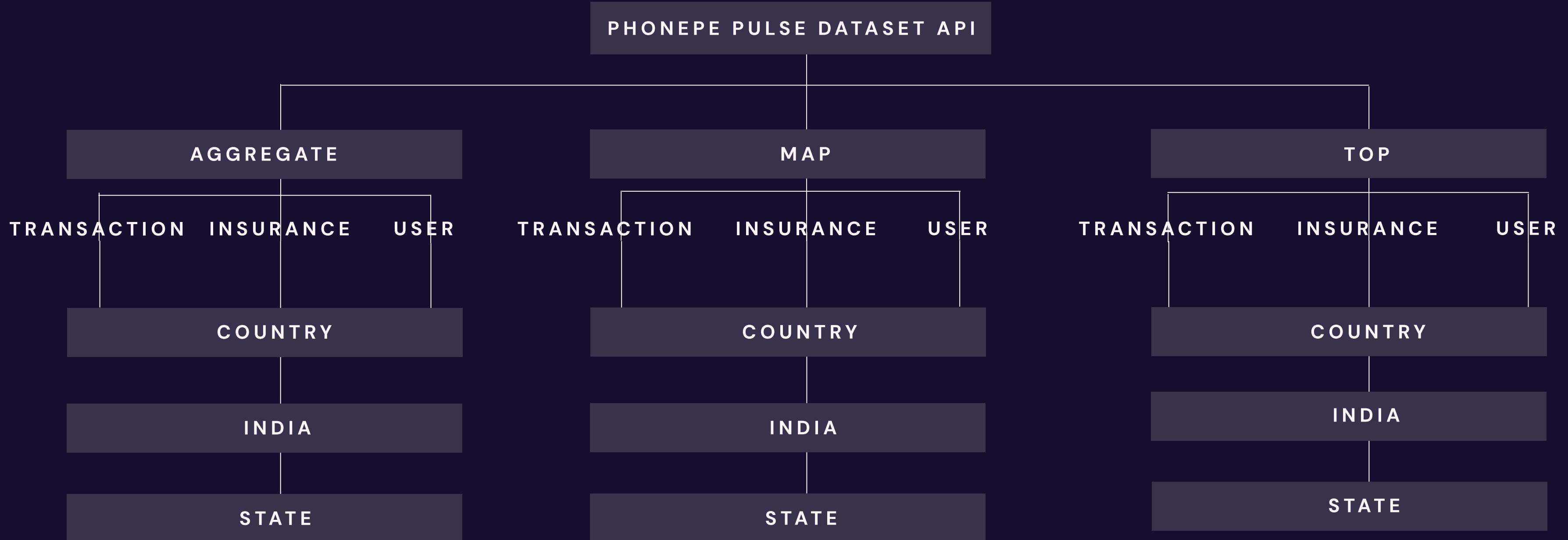
- Interactive dashboard displaying summary statistics, top 10 values for Transaction, Insurance, User domains.
- An interactive panel where users can filter data based on Year and Quarter and values will dynamically change.
- A chloropleth Indian map allowing users to hover over different states and get values accordingly.

## BUSINESS CASES

- Address five business case studies by providing key insights, analysis and business recommendations where applicable.

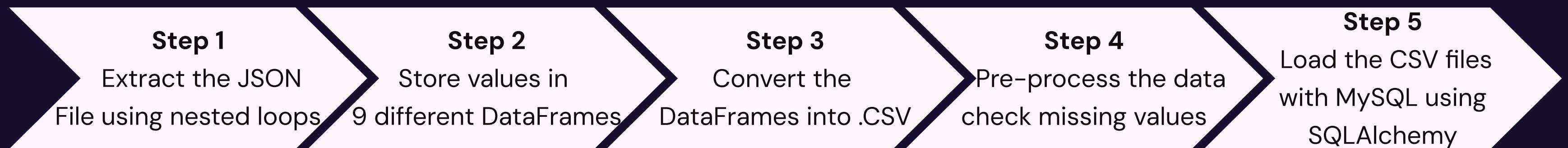
# PhonePe Pulse Dataset API

- The dataset was obtained from: <https://github.com/PhonePe/pulse>.
- It is in the form of a **nested JSON file** with **multiple folders** and its structure is as follows:



- For the SQL Queries running in the background, not all the tables were necessary to use.

# Data Extraction



**Note:** When the files were extracted, for each of **Aggregate, Map, Top (Transaction, Insurance, User)**, 9 different DataFrames were created by creating the columns and assigning values to those columns based on a **nested for loop**.

# MySQL Connection

- To create the connection with MySQL:
  1. PyMySQL was used for the MySQL Driver
  2. SQLAlchemy was used to generate the MySQL Database Engine
- This connection links MySQL to Jupyter Notebook and allows running and executing SQL queries in Python

```
try:  
    connection_string = f"mysql+pymysql://{username}:{password}@{host}:{port}/{database}"  
    engine = create_engine(connection_string) # creates an engine object  
    print ("Connected successfully to MySQL")  
  
    with engine.connect() as conn: # asks the database for a real engine connection and r  
        result = conn.execute(text("SELECT DATABASE();")) # executing a sample query to s  
        print ("Connected to: ", result.scalar())  
  
except Exception as e: # exception handling  
    print (f"Error connecting to the database: {e}")
```

```

# Metrics
if dataset_type == "Transactions": # User selects transaction
    total_txn_count = df["Transaction_count"].sum() # provides count of transactions
    total_txn_amount = df["Transaction_amount"].sum() # total transaction amount
    avg_txn_value = total_txn_amount / total_txn_count if total_txn_count else 0 # calculate average value
    # Create separate columns which will display aggregate values
    col1, col2, col3 = st.columns(3)
    col1.metric("All PhonePe Transactions", f"{total_txn_count:.0f}")
    col2.metric("Total Transaction Amount", f"₹{format(int(total_txn_amount/1e7),',')} Cr")
    col3.metric("Average Transaction Value", f"₹{avg_txn_value:.0f}")

elif dataset_type == "Insurance": # User selects Insurance
    total_ins_count = df["Insurance_count"].sum() # Insurance transaction count
    total_ins_amount = df["Insurance_amount"].sum() # Insurance amount
    avg_ins_value = total_ins_amount / total_ins_count if total_ins_count else 0 # Average insurance value
    # Create separate columns which will display aggregate values
    col1, col2, col3 = st.columns(3)
    col1.metric("All PhonePe Insurance Transactions", f"{total_ins_count:.0f}")
    col2.metric("Total Insurance Amount", f"₹{total_ins_amount/1e7:.0f} Cr")
    col3.metric("Average Insurance Amount", f"₹{avg_ins_value:.0f}")

else: # if user selects Users
    total_users = df["Registered_users"].sum() # number of registered users
    total_apps = df["Number_of_appOpens"].sum() # frequency of PhonePe app opens
    col1, col2 = st.columns(2)
    col1.metric("Total Registered Users", f"({total_users:.0f})")
    col2.metric("PhonePe App Opens", f"({total_apps:.0f})")

```

```

try:
    with engine.connect() as conn:
        years = pd.read_sql(f"SELECT DISTINCT Year FROM {map_table} ORDER BY Year;", conn)[["Year"]].tolist()
        quarters = pd.read_sql(f"SELECT DISTINCT Quarter FROM {map_table} ORDER BY Quarter;", conn)[["Quarter"]]
    selected_year = st.sidebar.selectbox("Select Year", years, index=len(years) - 1) # Here users can make the
    selected_quarter = st.sidebar.selectbox("Select Quarter", quarters) # Users can select the quarters
    query = text(f"""SELECT * FROM {map_table} WHERE Year = :year AND Quarter = :quarter""") # run the query by
    with engine.connect() as conn:
        df = pd.read_sql(query, conn, params={"year": selected_year, "quarter": selected_quarter})

    st.subheader(f"{dataset_type} Data Overview - {selected_year} Q{selected_quarter}") # Data Overview change.

if dataset_type == "Transactions":
    if agg_table:
        with engine.connect() as conn:
            payment_category_query = text(f"""SELECT Transaction_type AS Transaction_type, SUM(Transaction_amount)
                AS Total_Value
                FROM {agg_table}
                WHERE Year = :year AND Quarter = :quarter
                GROUP BY Transaction_type
                ORDER BY Total_Value DESC""")
            category_df = pd.read_sql(payment_category_query, conn, params={"year": selected_year,
                "quarter": selected_quarter})
            category_df["Total_Value"] = pd.to_numeric(category_df["Total_Value"].round(0), downcast="integer")
            fig = create_styled_table(category_df, "Transaction_type", "Total_Value", "Payment Categories", 120, 150)
            fig.update_layout(height = 250)
            st.plotly_chart(fig, use_container_width = True)

    else:
        st.warning("Aggregation table for Payment Categories not available.")

```

# Sample SQL Queries

- The SQL queries shared on the left are the ones used for creating the interactive dashboard on the 'EXPLORE DATA' Page where they have these placeholders for "Year", "Quarter" and based on the user's selection, the value will be input in the query and run accordingly.
- When these queries run, the values will automatically change on the dashboard. The following is a sample showing the values change when the year is changed from 2023 to 2024:

## Payment Categories

Transaction_type	Total_Value
Peer To Peer Payments	15,924,791,161,756
Merchant Payments	3,745,974,875,576
Recharge & Bill Payments	829,946,831,715
Others	7,571,850,963
Financial Services	5,673,295,497

## Payment Categories

Transaction_type	Total_Value
Peer To Peer Payments	22,327,372,608,425
Merchant Payments	5,962,408,968,716
Recharge & Bill Payments	1,150,328,145,543
Financial Services	11,806,137,694
Others	6,518,029,314

# Business Case Studies

1. Decoding Transaction Dynamics on PhonePe
2. Device Dominance and User Engagement Analysis
3. Insurance Penetration and Growth Potential Analysis
4. User Engagement and Growth Strategy
5. Insurance Transactions Analysis

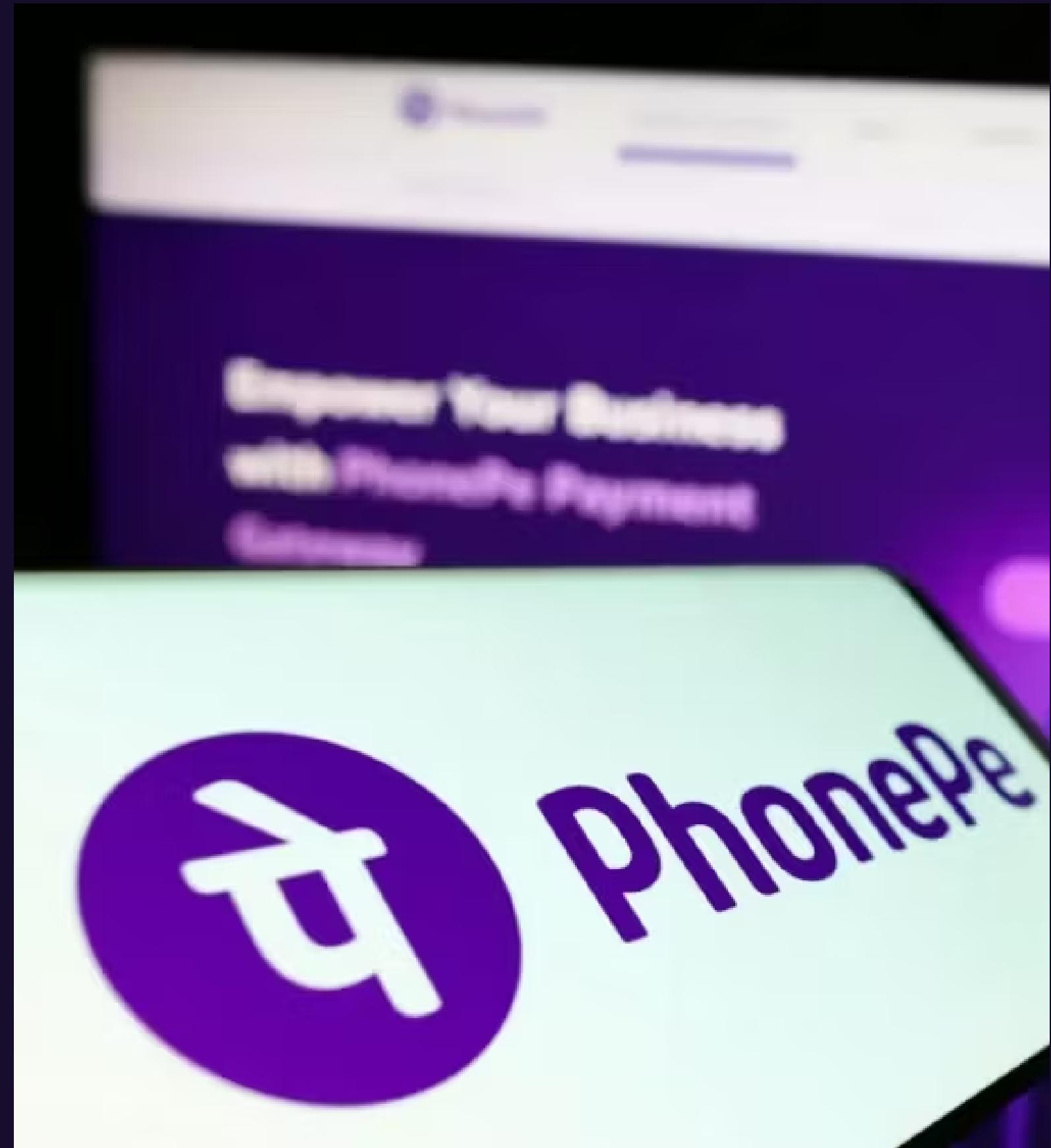
KEY INSIGHTS



ANALYSIS

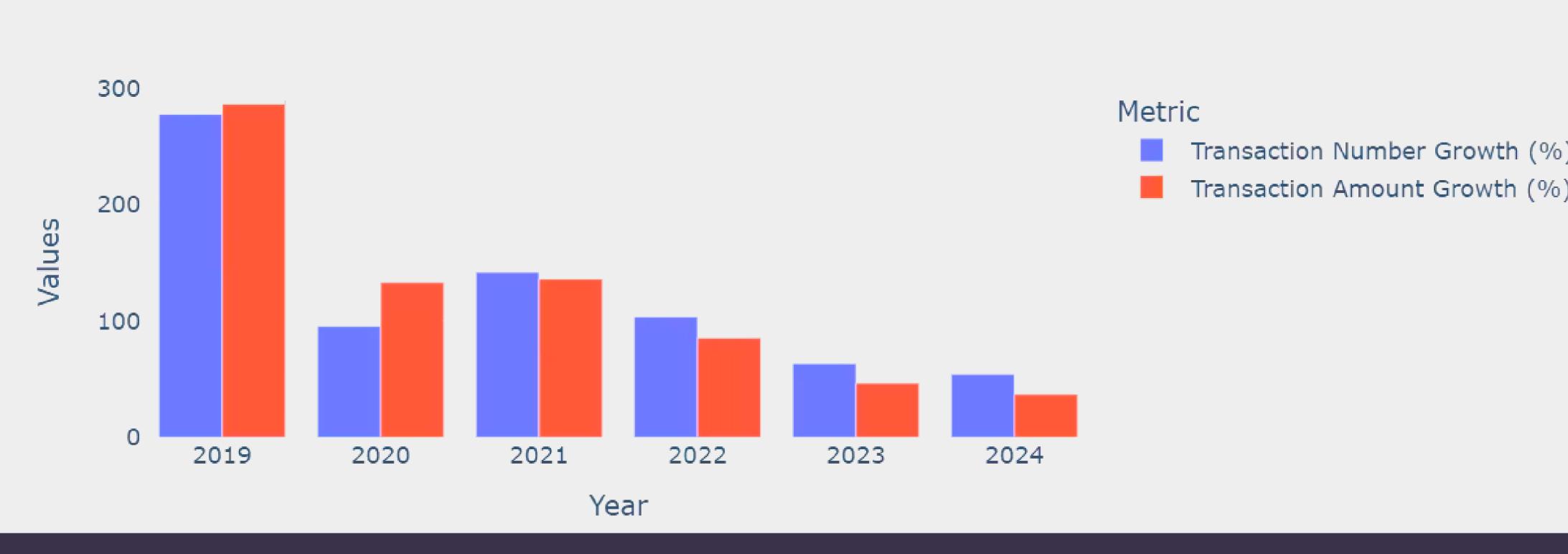


BUSINESS RECOMMENDATIONS



# Case Study 1

## 1. GROWTH IN PHONEPE'S TOTAL TRANSACTION VOLUME (2019-2024)



### KEY INSIGHTS

**223.15%**

Decrease in growth of transaction volume between 2019 and 2024.

### ANALYSIS

- Market saturation
- Competition with other digital payment apps like GooglePay, Paytm.

### RECOMMENDATIONS

- Launch tiered-transaction milestones with rewards.
- Build and market 2-3 signature Features.

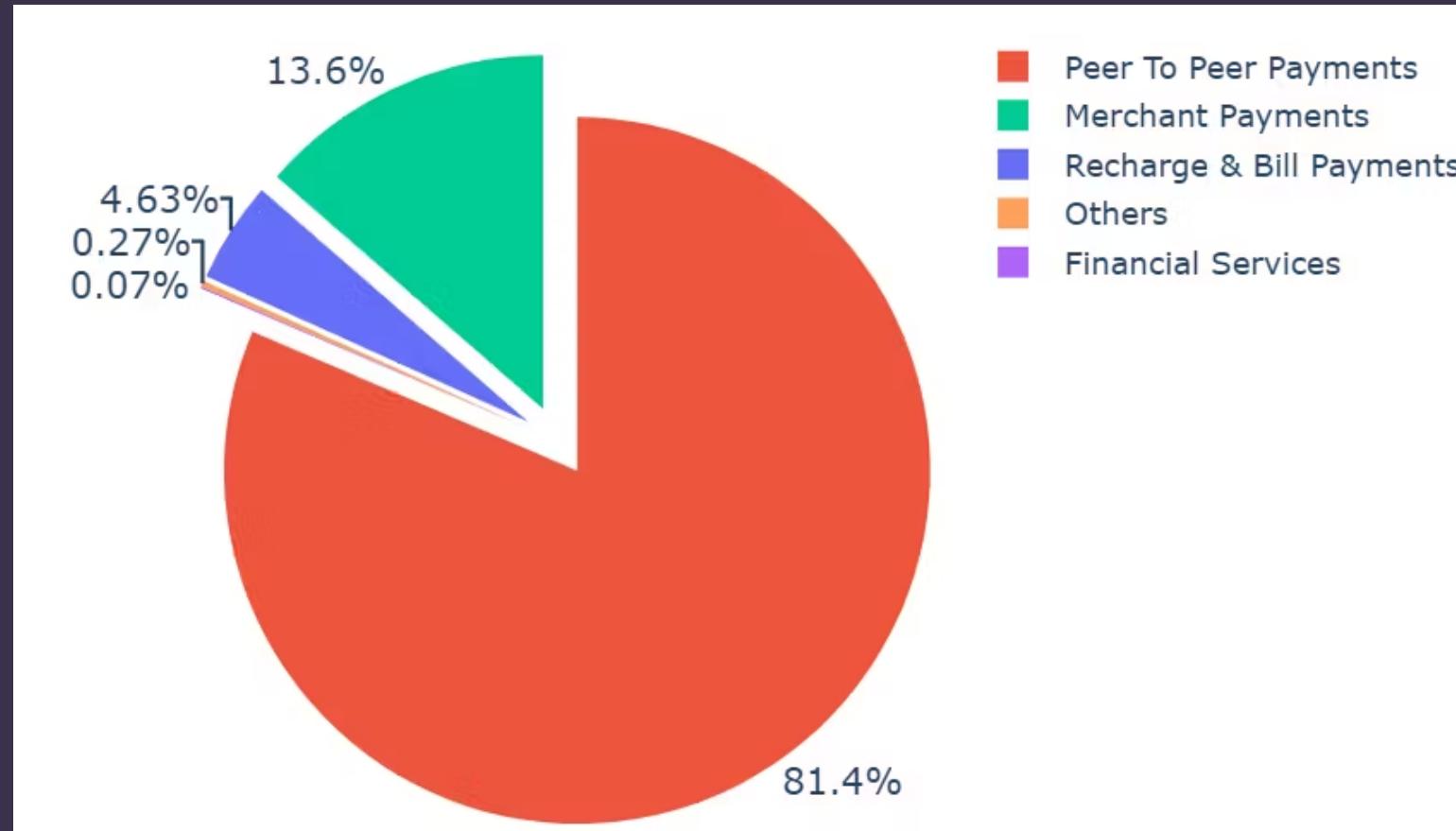
## Case Study 1

### DECODING TRANSACTION DYNAMICS ON PHONEPE

PhonePe has identified **significant variations** in transaction behavior across states, quarters, and payment categories. While some regions and transaction types demonstrate **consistent growth**, others show **stagnation or decline**.

The leadership team seek a deeper understanding of these patterns to drive targeted business strategies.

## 2. PERCENTAGE SHARE OF ALL TRANSACTIONS BY PAYMENT TYPE



### WHAT DO EACH PAYMENT TYPE MEAN?

- Peer To Peer Transactions** – Money transfer between 2 individuals using PhonePe wallet.  
(Eg: Paying Rent to Landlord).
- Merchant Payments** – Payments made to merchants for goods or services  
(Eg: Buying products from E-Commerce platforms).
- Recharge & Bill Payments** – Paying utility bills or recharging prepaid services (Eg: gas bills).
- Financial Services** – Purchases related to insurance or investment products (Eg: SIPs)

### KEY INSIGHTS

**81.4%**

of the total PhonePe payments are Peer to Peer while only **0.07%** are for Financial Services

### ANALYSIS

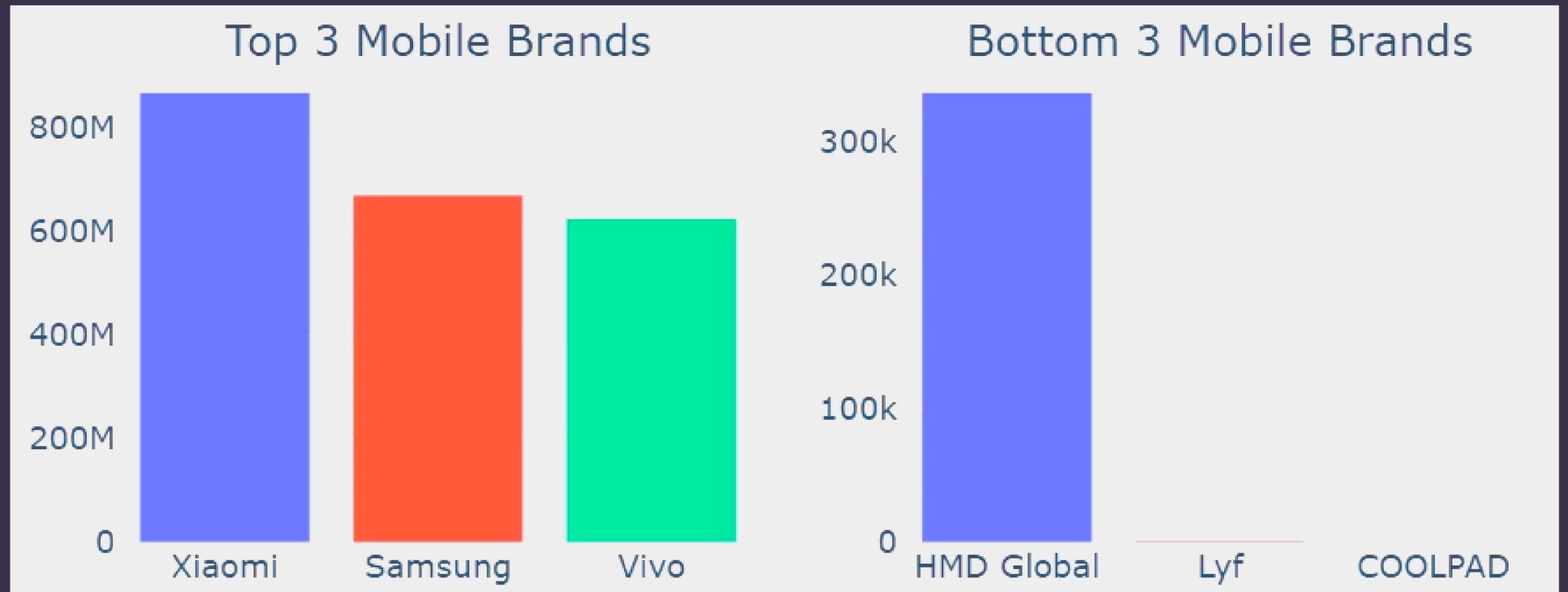
- P2P dominated PhonePe's early use for transactions among friends and family.**
- Low insurance awareness** limited financial services uptake.

### RECOMMENDATIONS

- Target QR saturation,** zero MDR onboarding, 5% cashback
- One-tap SIP/insurance nudges** after P2P Transaction.
- P2P-to-Pay hybrid:** auto bill split

# Case Study 2

## 1. MOBILE DEVICE BRANDS AND THEIR TOTAL NUMBER OF PHONEPE USERS



### KEY INSIGHTS

# 869.56M

Is the number of app users for Xiaomi which is the highest followed by Samsung and Vivo Lyf, COOLPAD the lowest.

### ANALYSIS

- **High market penetration** for brands like Xiaomi, Samsung and large user base compared to HMD Global, Lyf, COOLPAD.

### RECOMMENDATIONS

- Expand **pre-install deals** to grow user base for Top 3 brands.
- **Partner** with small brands via revenue-sharing deals.

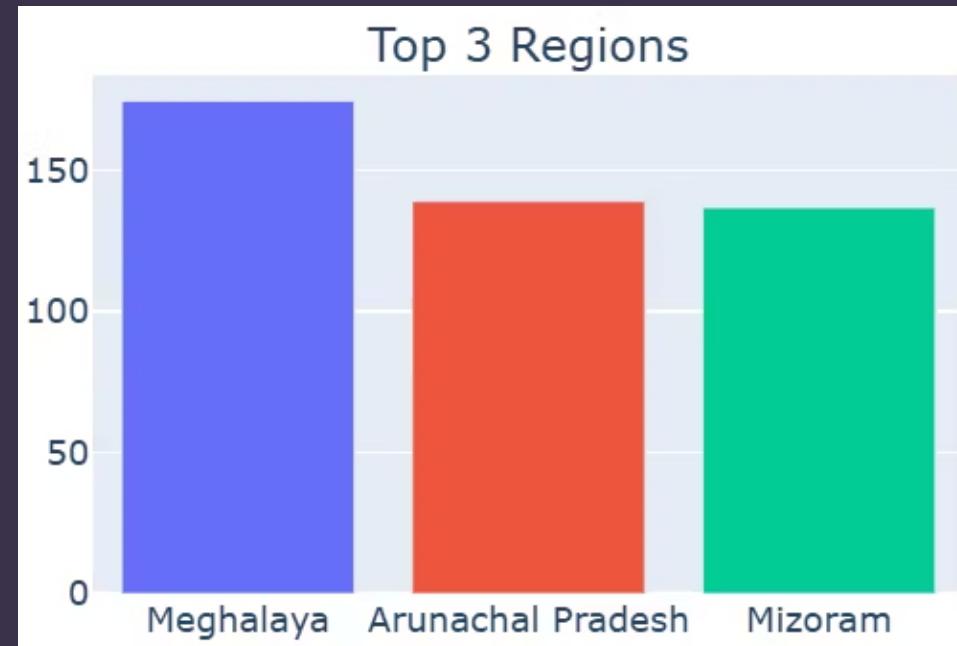
## Case Study 2

### DEVICE DOMINANCE AND USER ENGAGEMENT ANALYSIS

PhonePe seeks to **boost user engagement** and **optimize app performance** by analyzing user preferences across various device brands.

However, trends in device usage patterns differ widely between regions, with **certain devices** seeing **far less activity** than expected, despite high registration figures which needs to be addressed.

## 2. REGIONS WITH THE HIGHEST AND LOWEST APP ENGAGEMENT RATES



## WHAT DOES APP ENGAGEMENT RATE MEAN?

- It refers to the number of **registered users who actually open and use the PhonePe app** over a specific period of time.
- The rate is calculated by:  
$$\frac{\text{Average Number of App Opens (ao)}}{\text{Registered Users (r)}}$$

### KEY INSIGHTS

**174.36 ao/r**

Is Meghalaya's PhonePe app engagement rate making it the highest in the country followed by Arunachal Pradesh and Mizoram while Delhi (14.93) and Chandigarh (13.40) have the lowest.

### ANALYSIS

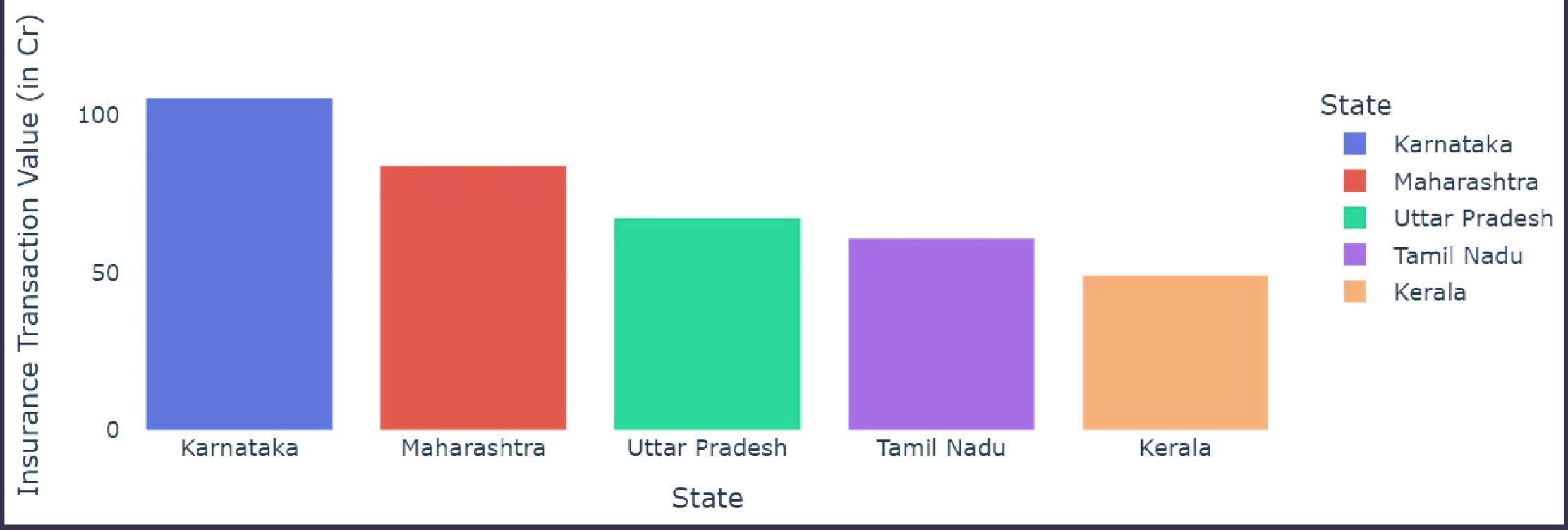
- Meghalaya, Arunachal Pradesh, Mizoram show **rapid smartphone adoption and digital payments**, relying heavily on apps like PhonePe for daily transactions, while Delhi and Chandigarh are established markets often facing **immense competition**.

### RECOMMENDATIONS

- Localized language packs** with voice-assisted UPI/offline QR mode to amplify Northeast growth.
- Run '**PhonePe Exclusive Contests**' in mature markets like Delhi providing 5% cashback in these contests and special metro perks at stores to boost app usage.

# Case Study 3

## 1. STATES WITH HIGHEST INSURANCE TRANSACTION VALUE OVER YEARS



### KEY INSIGHTS

**105 Cr**

Is the total insurance amount for Karnataka which is the highest in the country.

### ANALYSIS

- Karnataka is a state with **high financial literacy** and **tech-savvy** consumers especially Bengaluru.

### RECOMMENDATIONS

- **Target HNIs** in Karnataka and other states with **premium bundles**.
- Conduct “insurance literacy drives”.

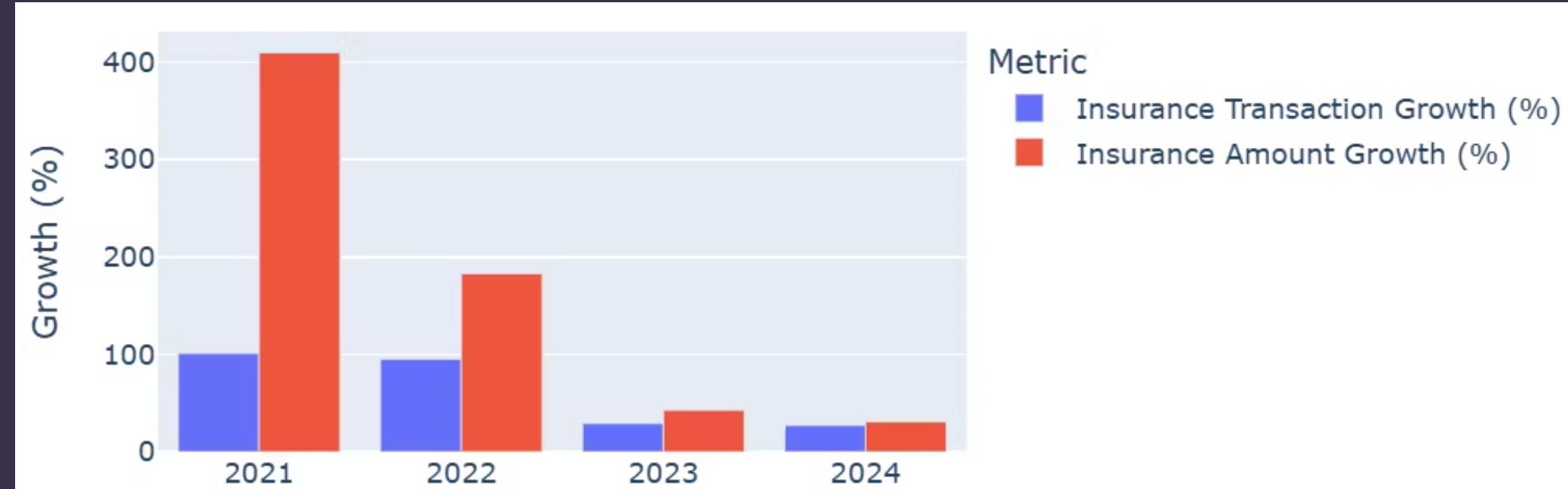
## Case Study 3

### INSURANCE PENETRATION AND GROWTH POTENTIAL ANALYSIS

PhonePe has expanded into the **insurance sector**, offering users a range of policy options for coverage.

As transactions in this area grow, the company aims to track its **development** and discover **new opportunities** for insurance adoption at the state level. Data will help **prioritize regions** for marketing efforts and **partnership with insurers**.

## 2. GROWTH IN NUMBER AND VALUE OF INSURANCE TRANSACTIONS (2021-2024)



### KEY INSIGHTS

**63%**

Decrease in insurance transaction growth between 2021 and 2024 after its peak in 2021 while the amount spent on insurance transactions decreased by **369%**

### ANALYSIS

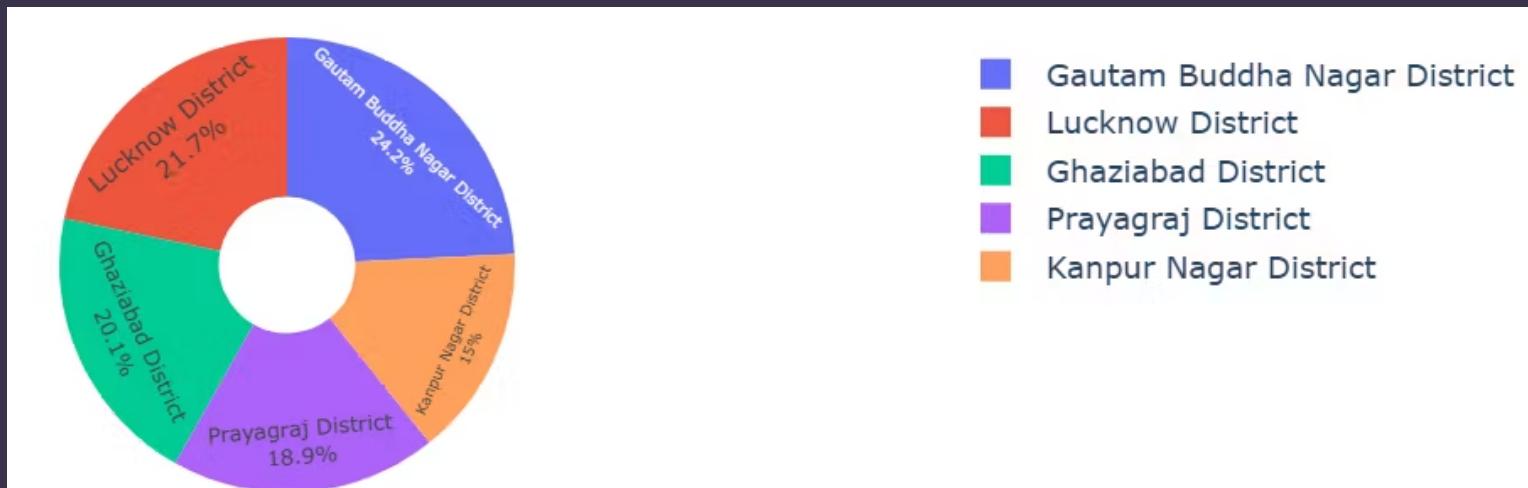
- The sharp spike in 2021 due to company's launch of **insurance products and aggressive expansion** into digital market.
- After 2021 surge, **adoption** plateaued along with competition in the digital insurance space slowing **subsequent growth**.

### RECOMMENDATIONS

- Reactivate lapsed insurance users by reaching out to 2021 buyers with **10-15% cashback** for renewing on PhonePe + **one-tap option** to upgrade.
- Send **smart renewal reminders** involving **Push notifications** 30 days before policies expire, paired with easy **UPI autopay setup**.
- Bring insurance products to **low-literacy segments** by rolling out **voice-guided insurance quizzes** in **regional languages** with added incentives or rewards.
- Partnership with **local kirana** networks for **online-offline funnel** thereby leveraging daily merchant visits by scanning QR at shops for instant term plans.

# Case Study 4

## 1. PERCENTAGE DISTRIBUTION OF APP OPENS ACROSS DISTRICTS WITHIN TOP 3 STATES WITH HIGHEST APP OPENS



UTTAR PRADESH



KARNATAKA



MAHARASHTRA

## Case Study 4

### USER ENGAGEMENT AND GROWTH STRATEGY

PhonePe aims to strengthen its market presence by examining **user activity patterns across various states and districts**.

By leveraging data on registered users and app usage, the company can uncover key behavioral trends to inform **strategic decisions** and drive **future growth**.

KEY INSIGHTS, ANALYSIS AND RECOMMENDATIONS ON THE NEXT SLIDE →

## KEY INSIGHTS

- Karnataka, Maharashtra and Uttar Pradesh have the highest number of PhonePe app opens across India.
- For Karnataka, Urban Bengaluru has the highest share of 59% followed by Belagavi district with 14.3%
- For Maharashtra, Pune dominates the shares with 41.6% followed by Nashik with a share of 22.5%.
- For Uttar Pradesh, Top 3 districts have similar shares with Gautam Buddha Nagar having the highest share of 24.2% followed by Lucknow (21.7%)

## ANALYSIS

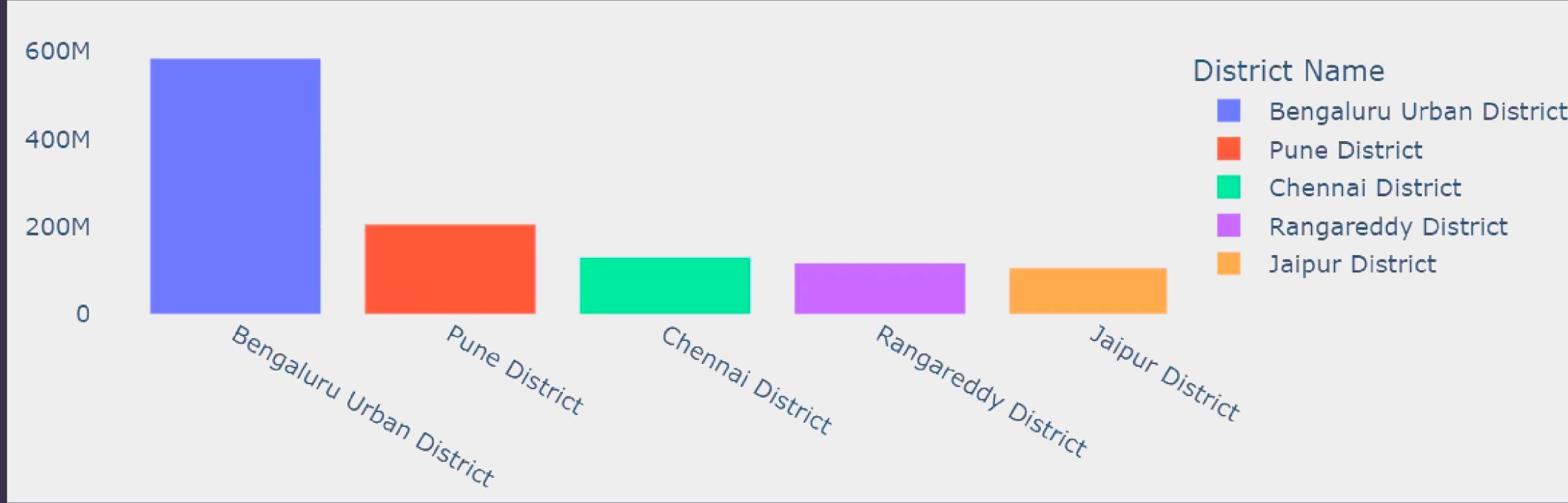
- The top districts in each state Pune, Nashik, Bengaluru, Lucknow, Ghaziabad and Gautam Buddha Nagar (which includes Noida and Greater Noida) are **major urban centers** with **significant economic activities** and **high disposable income** and a **tech-savvy population**.
- These places have **high merchant densities** making them a hub for merchant payments and retail transactions.
- Moreover, have dense population with **higher smartphone penetration** resulting in **higher PhonePe app engagement**.

## RECOMMENDATIONS

- Target Indian festivals in the top districts by creating a "**Merchant Onboarding Blitz**" where 10000+ new QR codes per district are installed as UPI peaks by 37% after 6 PM.
- Offer **urban premium packages** with **higher transaction limits** + exclusive rewards for these top districts.
- Create **strategic partnerships** with **tech ecosystems** such as IT parks, co-working spaces in Bengaluru/Pune for **exclusive PhonePe integrations**.
- PhonePe can also create **district leaderboards** within their apps to **gamify engagement**.

# Case Study 5

## 1. TOP 5 DISTRICTS WITH THE HIGHEST INSURANCE TRANSACTION VALUE IN 2024



### KEY INSIGHTS

**58.4Cr**

Is the total transaction value of Bengaluru Urban District which is the highest in 2024.

### ANALYSIS

- Bengaluru has a **large income tech-savvy** population who are **well-educated** and financially literate with high disposable incomes.
- Many companies in Bengaluru encourage or provide **insurance options** as part of **employment compensation** packages thereby increasing **awareness** and **uptake** of products.

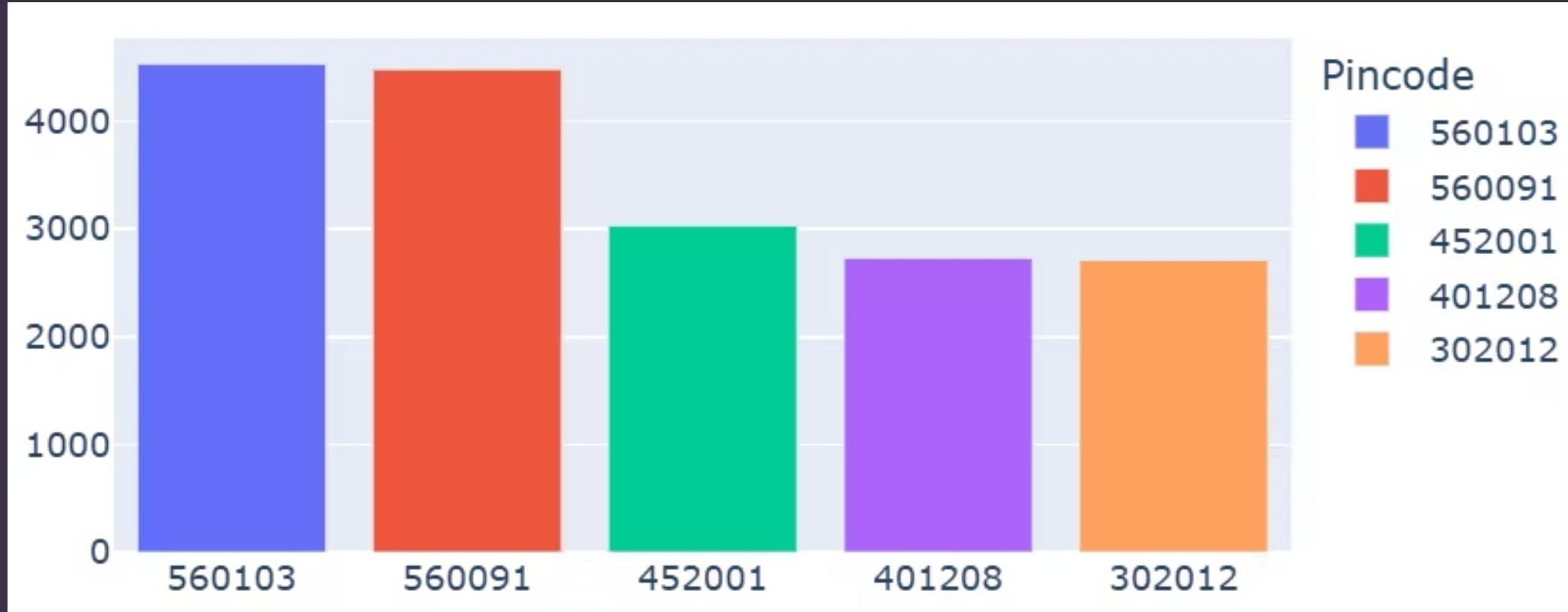
## Case Study 5

### INSURANCE TRANSACTIONS ANALYSIS

PhonePe seeks to examine insurance transactions to pinpoint the **leading states, districts, and pin codes** with the highest activity for a given year and quarter.

This evaluation will aid in gauging **user engagement** with **insurance products** and guide strategic decision-making.

## 2. TOP 5 POSTAL CODES WITH THE HIGHEST GROWTH IN INSURANCE TRANSACTION COUNT IN 2024



### KEY INSIGHTS

- Postal codes 560103, 560091, 452001, 401208, 302012 have the highest growth in number of insurance transactions in 2024.
- Within these postal codes, 560103 and 560091 display the highest growth with an increase of 4530 units compared to the other postal codes.

### ANALYSIS

- These postal codes on display belong to **prominent urban and suburban cities** like Bengaluru (560103, 560091), Indore (452001), Mumbai Suburban (401208) and Jaipur (302102).
- These regions have a **higher concentration of working professionals**, wealthier residents and a **strong digital adoption culture** fueling rapid insurance uptake through PhonePe.
- It is also likely that PhonePe actively **focused its marketing and outreach efforts** in these postal codes, tapping into neighborhoods known for early tech adoption.

# Streamlit Deployment

- Once the **queries** and **figures** were ready, along with the business case studies, the **multi-page interactive dashboard** was created using **Streamlit** where users can navigate through 3 different pages: **HOME, EXPLORE DATA, BUSINESS CASES** and use the filters to view data accordingly.
- The Streamlit dashboard was created within **Jupyter Notebook** but in order for it to be viewed by other users, it needed to be hosted online. To accomplish this, the following steps were carried out:

## 1. GITHUB

- The following files were uploaded on **GitHub**: **PhonePe\_Logo.png**, **app.py** (consisting of all the SQL queries, figures, Streamlit dashboard creation code), **requirements.txt** (required libraries and packages to import), **agg/map/top tables** (total 9 .csv files) for **version control** and **automated deployment**.



## 2. RAILWAY MYSQL

- A **external production MySQL database** was created on **Railway** due to limitations within the **Streamlit Community Cloud platform** to allow **built-in database hosting**.
- GitHub credentials** were used to login, keeping the **app.py file as the main source** file and Railway automatically generated connection credentials (**host, port, username, password, database name**)

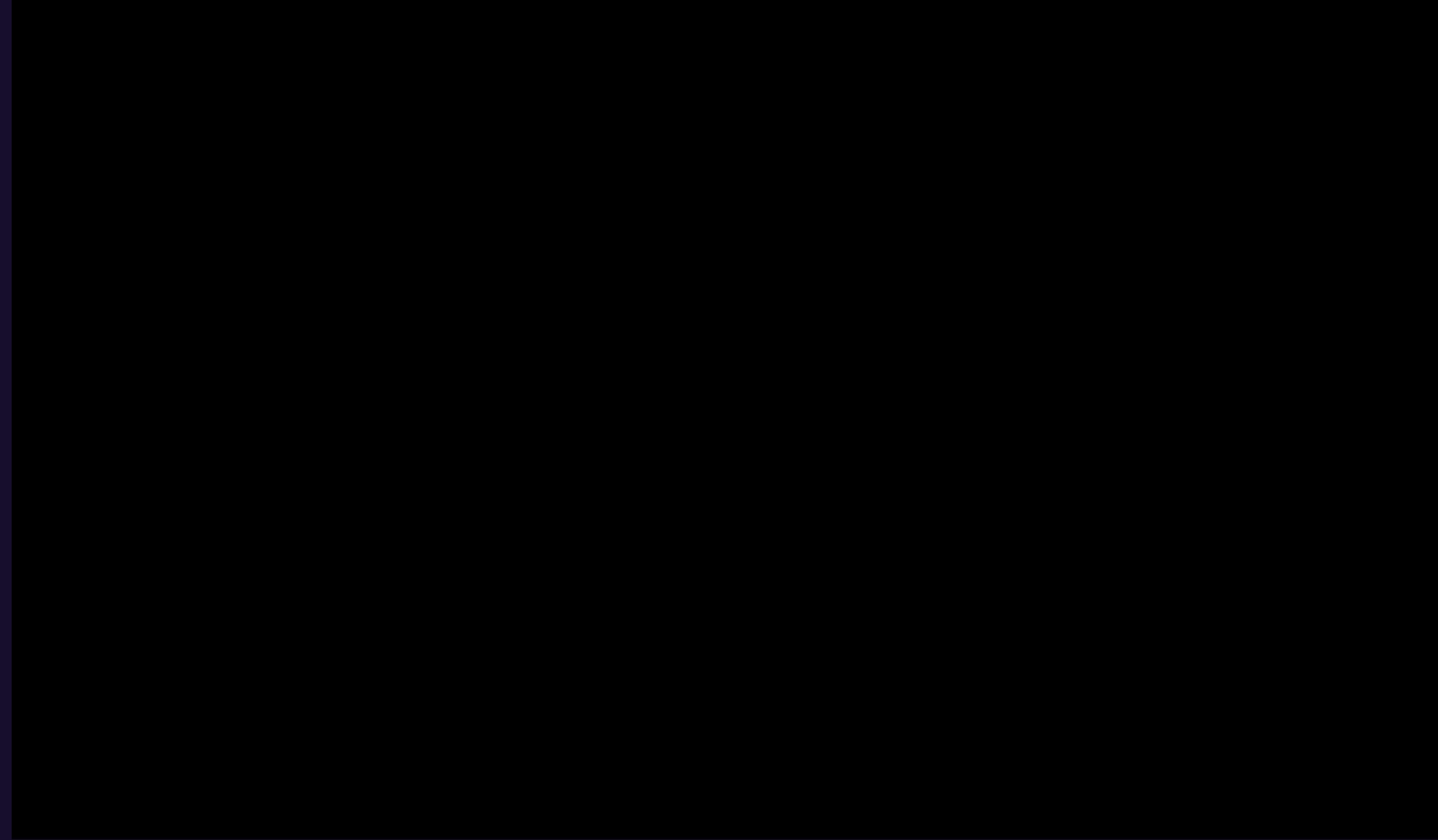


## 3. STREAMLIT COMMUNITY CLOUD

- The Railway connection credentials were then pasted in the **Streamlit Community Cloud Secrets** which used them to connect to the app.py in GitHub and deploy the interactive dashboard online.



# PhonePe Pulse Dashboard Video



[https://phonepepulsebusinessanalyticsproject-  
yxp3o38574xaw8mj9pkrym.streamlit.app/](https://phonepepulsebusinessanalyticsproject-yxp3o38574xaw8mj9pkrym.streamlit.app/)